

ABSTRACT

Title of thesis: HUMAN APPEARANCE MODELING
IN VISUAL SURVEILLANCE

Yang Yu, Master of Science, 2007

Thesis directed by: Professor Rama Chellapa
Department of Electrical and Computer Engineering

We present an appearance model for establishing correspondence between tracks of people which may be taken at different places, at different times or across different cameras.

Illumination insensitive color features, i.e., RGB rank feature and brightness-color feature are used. Path-length feature is added for structural information and invariance to motion and pose. The appearance model is constructed by kernel density estimation. Kullback-Leibler distance measures the similarity between the models. To further exploit the information in video sequence, key frame selection method and online hierarchical clustering algorithm are proposed to construct appearance model from video. Key frame selection use the frames with large information gain to represent the appearance model. Online hierarchical clustering algorithm condense the model into a few clusters in the framework of our appearance model.

Experimental results demonstrate the important role of the path-length feature in the appearance model and the effectiveness of the proposed appearance model and matching method.

HUMAN APPEARANCE MODELING IN VISUAL SURVEILLANCE

by

Yang Yu

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2007

Advisory Committee:
Professor Rama Chellapa, Chair/Advisor
Professor Larry Davis
Dr. David Doermann

© Copyright by
Yang Yu
2007

Table of Contents

List of Tables	iv
List of Figures	v
1 Introduction	1
2 Human Appearance Modeling from Snapshot	5
2.1 Overview	5
2.2 Review of existing human appearance models	6
2.3 Color path-length profile	7
2.3.1 Path-length	8
2.3.2 Color feature	10
2.3.2.1 RGB Color	10
2.3.2.2 Normalized RGB Color	11
2.3.2.3 RGB Rank	11
2.3.2.4 Brightness and Color Decomposition	13
2.4 Appearance Model	14
2.4.1 Appearance Model Using Kernel Density Estimation	14
2.4.2 Matching of Appearances	16
2.5 Evaluation	19
2.5.1 Bandwidth Selection	21
2.5.2 Experiment Setting	24
2.5.3 Study of Color-Pathlength Profile	25
2.5.4 Study of Scale and Subsampling	29
2.5.5 Local Appearance Difference	32
2.6 Conclusion	32
3 Human Appearance Modeling of Video Sequence	34
3.1 Overview	34
3.2 Key Frame Selection	35
3.2.1 Algorithm	35
3.2.2 Experiment Results	37
3.3 Online Clustering	42
3.3.1 Algorithm	43
3.3.2 Experiment Results	48
3.4 Online Hierarchical Clustering	48
3.4.1 Algorithm	51
3.4.2 Experiment Results	56
3.5 Conclusion	65
4 Conclusion	69

List of Tables

2.1	Algorithm of Computing Path-Length	10
2.2	Bandwidth selection for R,G,B color and pathlength	19
2.3	Bandwidth selection for brightness and path-length	20
2.4	Optimal Bandwidth of different features	24
2.5	Performances of different features when snapshots of the two cameras are of similiar pose	25
2.6	Performance of different sequence sets. Feature of RGB rank and path-length is used	29
2.7	Performance of matching when snapshots of camera 1 are scaled. Feature of RGB rank and path-length is used	30
2.8	Matching result of the sequence set of different illumination same pose when sub-sampling is applied. Feature of RGB rank and path-length is used	31
3.1	Matching results of Honeywell sequences when the proposed key frame selection and matching scheme is used	37
3.2	Matching results of outdoor sequences when the proposed key frame selection and matching scheme is used	38
3.3	A Classic Online Clustering Algorithm	44
3.4	The Proposed Cluster Center Update Algorithm	46
3.5	Proposed Online Clustering Algorithm	47
3.6	Agglomerative Hierarchical Clustering Algorithm	52
3.7	Algorithm of Merging Clusters	53
3.8	Hierarchical Clustering Algorithm for Appearance Modeling	67
3.9	Matching results when appearance models are built using hierarchical clustering	68

List of Figures

2.1	Brightness distance and color distance between feature value \mathbf{x} and the sample pixel value \mathbf{x}_i	12
2.2	Dataset Honeywell: Indoor image sequences taken by two cameras collected by the Honeywell corporation	23
2.3	CMC of different features when snapshots of the two cameras are of similiar pose	26
2.4	Using a large brightness bandwidth may confuse bright color with dark color	28
2.5	The log-likelihood ratio image (c) reflects the local appearance difference between the test image (a) and the model image (b). Here a brighter pixel indicates a larger log-likelihood ratio	31
3.1	Indoor sequence: example result of key frame selection	39
3.2	Dataset 2: Outdoor image sequences	40
3.3	Outdoor sequence: example result of key frame selection	41
3.4	Online clustering result of sequence A	49
3.5	Online clustering result of sequence B	50
3.6	Level 1 of hierarchical clustering result of figure 3.4	58
3.7	Level 2 of hierarchical clustering result of figure 3.4	59
3.8	Level 3 of hierarchical clustering result of figure 3.4	60
3.9	Level 1 of hierarchical clustering results of figure 3.5	61
3.10	Level 2 of hierarchical clustering results of figure 3.5	62
3.11	Level 3 of hierarchical clustering results of figure 3.5	63
3.12	Level 4 of hierarchical clustering results of figure 3.5	64

Chapter 1

Introduction

Nowadays visual surveillance systems are widely deployed for security purpose in public places such as a subway, airport, or parking lot. While in traditional visual surveillance systems video data are directly interpreted by humans, there is growing interest in developing automated video understanding techniques to both reduce the cost of the system and relieve the burden of tedious analysis by humans. These techniques automate the analysis of video data in a variety of settings and configurations. For example, in the simplest case, cameras are stationary and moving objects are to be detected, which usually requires background subtraction. More elaborate applications such as tracking try to follow an object through a sequence by analyzing the sequence of video frames. The most sophisticated scenarios involve interpreting the event in the video such as leaving an unattended luggage, theft or violence.

Different surveillance systems have different configurations. In a basic visual surveillance system, only a single camera is mounted to cover the scene of interest. Some visual surveillance systems have multiple cameras observing the same scene at different angles. Finally, in systems of wide areas such as airports, highways or shopping centers multiple cameras are distributed to cover the area under surveillance. To

automatically understand the activities of the surveillance systems, which is the ultimate goal of automated visual surveillance, one of the important sub-problems is to establish correspondence between observations of people who might appear, disappear and reappear at different times, within different scenes or across different cameras. For instance, one person may first appear in the entrance of a building, and then later the same person may appear in the hallway of the building, finally the person may exit from the entrance of the building. To know the activity of the person, such as when the person enters, where he/she has been to, and whether he/she leaves with a baggage, we should be able to know whether the observed person in the different cameras at different places is in fact the same individual.

The objective of this thesis is matching of a person, as it moves within different scenes, at different times or across different cameras. In most surveillance systems the appearance of a person does not change very much in spite of the temporal and spatial separation between observations from possibly different cameras. For example, people may first enter a building, then after a short time they may exit wearing the same clothing; or people may walk toward a camera at one end of a hallway, then toward another camera at the other end of the hallway with little appearance change. So in this thesis appearance feature is utilized to solve the problem. Although the appearance of people does not change very much between observations, there are still large variations induced by various factors, including the illumination conditions being different from cameras in different places, appearance variations caused by changing body postures and views, etc. Thus a successful

matching approach must have effective representations of appearance to accommodate the variations caused by illumination, pose and view changes.

This thesis aims to address the above appearance matching problem. The contribution of the thesis is summarized as follows:

1. The illumination invariance of color feature is studied. RGB rank feature and brightness-color feature are shown to be illumination invariant. Path-length feature is proposed to use together with the color feature. The path-length feature provides the structure information and is invariant to human poses and motions.
2. Kernel density estimation is employed to construct the probability model of human appearances. Then Kullback-Leibler distance, which measures the information gain, is used to do matching.
3. To extract the information in video sequence to further elaborate the appearance of people in video, a key frame selection algorithm is proposed. The key frame is defined to be the frame with large information gain to the previous key frame. A matching algorithm between the key frames of different people is proposed.
4. An online hierarchical clustering algorithm is proposed to construct the appearance model of people in video. Unlike the traditional hierarchical clustering algorithm, the data to be clustered are composed of feature samples which

cannot be described by simple parametric models. The merging algorithm and distance measure between clusters in this case are discussed and proposed.

The organization of the thesis is as follows. In chapter 2 the basics of the proposed appearance model are discussed. This includes the feature used, the distribution estimation method and distance measure employed. The effectiveness of the model is proved by matching between snapshots. In chapter 3 the construction of appearance model from video is discussed. The algorithms of key frame selection and online hierarchical clustering are proposed, and their experimental results are demonstrated. Finally chapter 4 summarizes and concludes the thesis.

Chapter 2

Human Appearance Modeling from Snapshot

2.1 Overview

This chapter presents our human appearance model, which is the basis of the thesis. Our appearance model is based on spatial /color statistical features. The color and path-length features of the pixels inside the silhouette of a person are used to construct an appearance model. Path-length, the length of the shortest path from a distinguished point, which is the top of the head here, to a point constrained to lie entirely within the body, captures the structural information of appearance. It has the property of an inner-distance [19], so is invariant to 2D-articulations. This makes it less sensitive to human motion than the spatial positions of the features. To cope with illumination change, color features robust to illumination change are combined with path-length to build the appearance model. We consider the illumination insensitive color features and path-length to be probabilistic random variables and estimate the color path-length distribution using kernel density estimation. Once the appearance model is built, correspondence between observations are found by minimizing the Kullback-Leibler distance, which measures the information gain between observations. We will show how the Kullback-Leibler distance is obtained when the probability density is approximated with kernel density estimation. Experimental results are demonstrated to show the effectiveness of our model.

2.2 Review of existing human appearance models

The most common appearance model is the color histogram [3, 8]. Although histogram-based approaches are very flexible and robust to non-rigid deformations, they do not contain any geometric appearance information. So, they cannot discriminate appearances that are the same in color distribution, but different in color structure. For example, these approaches cannot differentiate a person wearing a brown shirt and blue pants from a person wearing a blue shirt and brown pants. To achieve illumination invariance, [14] proposed to learn the brightness transfer functions across cameras through a training sequence and match appearances according to the probability of the color and the space and time relationship among the cameras.

To incorporate structure information, [6] proposed to use a joint feature-spatial space, where both the feature values and the spatial position of the features are taken as probabilistic random variables. Although this approach can discriminate differences due to structure, it is very sensitive to pose. For example, a walking person with left foot down and right foot up will be different from the same walking person with right foot down and left foot up. A person walking from left to right is different from the same person walking from right to left. So appearance features that are invariant to human pose are preferable. [18] proposed applying functionals over appropriate geometric sets for appearance modeling, which they refer to as *geometric transforms*. If a geometric transform is applied to different parts of the

human body, then a pose invariant appearance model can be obtained. Their approach requires automatically decomposing a human body into natural parts, which is itself a very difficult problem.

Other appearance models have been proposed for face recognition and vehicle matching applications. [24] measured the similarity between face sequences by the principle angle between the subspace spanned by the sequences. However the linear subspace assumption does not apply to human appearance since the local deformations resulting from motion are quite complicated. [22] proposed to first align the edges of vehicles and use the alignment features to match vehicles. It would be challenging to apply this to human appearance since wrinkles of clothing will result in many edges. [21] use shapeme histograms to construct and match vehicle representation from image sequences.

2.3 Color path-length profile

As we have pointed out in chapter 1, silhouettes of moving people are obtained by background subtraction and the tracks of each person have been generated. The background subtraction results have local errors (typically at the true boundaries of silhouettes) and occasional “catastrophic” failures, short subsequences of highly erroneous segmentation. Also as we have mentioned, we assume that the actual appearance of a person does not change very much between observations.

The ideal appearance feature should both easily discriminate different appearances and tolerate changes due to factors such as motion and illumination. The features we choose here are color and path-length of the pixels inside the silhouette of a person.

2.3.1 Path-length

The path-length of the pixel inside the silhouette is the length of the shortest path from a distinguished point, which we choose as the top of the head, to the pixel. A similar concept to path-length is inner-distance [19] which is defined as the shortest path between landmark points within the silhouette. In [19] it is shown that the inner-distance feature captures shape information and is insensitive to articulation. The path-length feature has a similar property. It not only reflects structure information, but is also insensitive to human motion which can be approximated as articulation. Due to the fact that the human body and clothing are typically bilaterally symmetric, the path-length is also invariant to poses that are bilaterally symmetric. For example, the color path-length feature of the front view of a walking person with left foot up and right foot down is very close to that of the same walking person with right foot up and left foot down. The color path-length feature of the side view of a person walking from left to right is close to the side view of the same person walking from right to left. Instead of using the centroid of the silhouette as the distinguished point [15] we choose the top of the head as the base point. The top of the head is easy to detect and relatively stable to movement.

Compared with the centroid of silhouette, it can discriminate the features of upper body and lower body because they have different path-lengths and do not produce mixed distributions. Finally, the top of the head is less sensitive to noise than the foot point, which can be hard to detect due to shadows.

In [11] the head point is predicted using the major axis of the silhouette, the hull vertices, and the topology of the estimated body posture. However, we found that the topmost point of the silhouette is the head point in most cases. So the head point of each segmented person is located as the middle point of the topmost row of the silhouette. Although this simple method could be wrong when there are some parts above the head point, it is usually correct with ordinary postures. Also as we will see later, our appearance model is a statistical model, which means each feature happens with a probability. This does not require an accurate path-length for each feature, or in other words, the head point is not necessarily accurate.

The simplified Dijkstra algorithm is used to find the path-length of each point in the silhouette to the head point. A queue is first built with the head point being the head of the queue. When each element of the queue is dequeued, the points that have not been visited in the neighborhood of the element are inserted into the queue. Also the distance to the head point, or the path-length, which is actually the distance to the element plus the path-length of the element is inserted into the queue at the same time. The psuedo-code of the algorithm is shown in table 2.1

Table 2.1: Algorithm of Computing Path-Length

Algorithm: Path-length Computation

```

 $Q$ .insert(head point)

Path_length(head point)  $\leftarrow$  0

while  $Q$  is not empty

     $p \leftarrow Q$ .removeHead()

    for  $q \leftarrow$  each neibgorhood of  $p$ 

        Path_length( $q$ )  $\leftarrow$  Path_length( $p$ ) + Distance( $p, q$ )

         $Q$ .AddTail( $q$ )

    end

end

```

2.3.2 Color feature

As the distance of people to the camera changes, or people move between different cameras, or get into different places such as in the shade or under the sun, the illumination changes, thus the values of the colors may change. To obtain correct matches of appearances, the feature should be invariant to illumination. Here we will discuss different color features that can be used in human appearance modeling.

2.3.2.1 RGB Color

The most commonly used three color components, RED, GREEN, and BLUE can be used directly. However, the values of RGB are greatly influenced by illumi-

nation, which results in incorrect matches.

2.3.2.2 Normalized RGB Color

Normalized RGB is formed independently from varying lighting levels. The red, green, and blue components of normalized RGB space can be obtained from the three components of RGB space using the following formulation:

$$r = \frac{R}{R + G + B} \quad (2.1)$$

$$g = \frac{G}{R + G + B} \quad (2.2)$$

$$b = \frac{B}{R + G + B} \quad (2.3)$$

Since $r + g + b = 1$, the normalized RGB color can also be represented by a brightness component y , and two color components r and g

$$y = R + G + B \quad (2.4)$$

$$r = \frac{R}{R + G + B} \quad (2.5)$$

$$g = \frac{G}{R + G + B} \quad (2.6)$$

In this way the color r and g are independent of illumination to some extent [20].

2.3.2.3 RGB Rank

Ranked color feature is based on the assumption that the shape of the color distribution function does not change very much under different illumination, so the percentage of image points of object i with color value less than \mathbf{x}_i is equal to the percentage of image points of the same object i in a different illumination condition

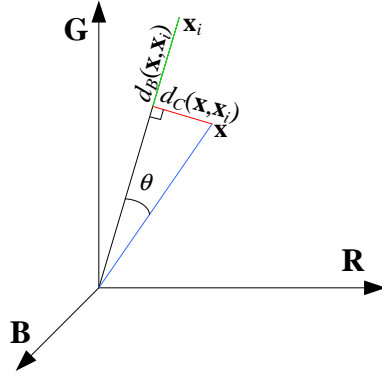


Figure 2.1: Brightness distance and color distance between feature value \mathbf{x} and the sample pixel value \mathbf{x}_i

with color value less than \mathbf{x}_j [10]. Ranked color feature disregard the absolute values of the color features and reflect the relative values instead. The ranked color features are invariant to monotonic color transforms, so are unchanged under a wide range of illumination changes.

In order to obtain the rank, the cumulative histogram H of the snapshot is first obtained. The rank $O(\mathbf{x})$ of feature value \mathbf{x} is the percentage value of the cumulative histogram

$$O(\mathbf{x}) = \lceil H(\mathbf{x}) \cdot 100 \rceil \quad (2.7)$$

where $\lceil x \rceil$ is the largest integer that is less than x .

2.3.2.4 Brightness and Color Decomposition

Decomposition of color into brightness and color component is based on the observation of a experiment did in [16]. In this experiment, a color chart is illuminated by a light that can adjust brightness. Different pixels in the color chart have different colors. As the illumination level of the light changes, the R, G, B values of the same pixel in the color chart are distributed in a cylinder whose axis goes toward the origin point of the RGB space and different pixels occupies different cylinders. In other words, illumination level change makes the R, G, B value move along the axis of the cylinder, and color change makes the R, G, B value move radially away from the axis of the cylinder. Based on this observation, we decompose the distance of the feature value to a sample pixel value into a brightness component and a color component as shown in figure 2.1. So the brightness and color component can be obtained as follows

$$\begin{aligned} d_B^2(\mathbf{x}, \mathbf{x}_i) &= (\|\mathbf{x}_i\| - \|\mathbf{x}\| \cos \theta)^2 \\ &= \left(\|\mathbf{x}_i\| - \frac{\langle \mathbf{x}, \mathbf{x}_i \rangle}{\|\mathbf{x}_i\|} \right)^2 \end{aligned} \quad (2.8)$$

and

$$\begin{aligned} d_C^2(\mathbf{x}, \mathbf{x}_i) &= \|\mathbf{x}\|^2 - \|\mathbf{x}\|^2 \cos^2 \theta \\ &= \|\mathbf{x}\|^2 - \left(\frac{\langle \mathbf{x}, \mathbf{x}_i \rangle}{\|\mathbf{x}_i\|} \right)^2 \end{aligned} \quad (2.9)$$

where $\|\mathbf{x}\|^2 = R^2 + G^2 + B^2$, $\|\mathbf{x}_i\|^2 = R_i^2 + G_i^2 + B_i^2$ and $\langle \mathbf{x}, \mathbf{x}_i \rangle = R \cdot R_i + G \cdot G_i + B \cdot B_i$.

2.4 Appearance Model

In this section we will first discuss the construction of the statistical appearance model using kernel density estimation. Then Kullback-Leibler distance is used to match the appearances.

2.4.1 Appearance Model Using Kernel Density Estimation

Statistical modeling is a useful tool to represent appearances. In statistical modeling, feature values are modeled as random variables in feature space with an associated probability density distribution. There are two types of statistical models, parametric models and nonparametric models. In parametric models, a specified statistical distribution is assumed to approximate the actual distribution and the associated parameters are estimated from the data. Gaussian model is the most commonly used parametric model. More complex parametric models involve multiple Gaussians, i.e., Gaussian mixture model. Alternatively, nonparametric models estimate the density function directly from the data without any assumptions about the underlying distribution, that is, selection of a model is avoided and estimation of the parameters is not needed.

The human appearances are so complex that it is very hard to describe using a specific distribution with several parameters. So in this thesis nonparametric model is employed. Particularly kernel density estimation (KDE) is used to estimate the

underlying density. In KDE, the underlying pdf is estimated as

$$f(x) = \sum_i \alpha_i K(x - x_i) \quad (2.10)$$

where K is the kernel function which typically is a Gaussian centered at the data points in feature space, $x_i, i = 1 \dots n$, and α_i are weighting coefficients. Formula (2.10) is actually estimating the p.d.f by averaging the effect of a set of kernel functions centered at each data point. It asymptotically converge to any density function with sufficient samples [4].

As we have discussed in the above section, color path-length profile is used as the feature to build the appearance model. Each pixel inside the silhouette of a snapshot is represented by the feature vector (\mathbf{x}, l) where \mathbf{x} is the feature value or color of the pixel and l is the path-length of the pixel. To achieve invariance to the size of image, the path-length feature is normalized by the height of the silhouette. Given all the pixels of a snapshot of a person, we estimate the distribution or p.d.f $p(\mathbf{x}, l)$, of the feature vector by kernel density estimation

$$p(\mathbf{x}, l) = \frac{1}{N} \sum_{i=1}^N k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{\mathbf{h}_{\mathbf{x}}}\right\|^2\right) k\left(\left\|\frac{l - l_i}{\sigma_l}\right\|^2\right) \quad (2.11)$$

where $k(\cdot)$ is the kernel function, and $\mathbf{h}_{\mathbf{x}}$ and σ_l are bandwidths of the feature value and path-length respectively. \mathbf{x} is the color feature. For example, if RGB color feature is used,

$$p(\mathbf{x}, l) = \frac{1}{N} \sum_{i=1}^N k\left(\left\|\frac{r - r_i}{h_r}\right\|^2\right) k\left(\left\|\frac{g - g_i}{h_g}\right\|^2\right) k\left(\left\|\frac{b - b_i}{h_b}\right\|^2\right) k\left(\left\|\frac{l - l_i}{\sigma_l}\right\|^2\right) \quad (2.12)$$

If brightness and color feature is used as described in section 2.3.2.4, the appearance model is

$$p(\mathbf{x}, l) = \frac{1}{N} \sum_{i=1}^N k \left(\left\| \frac{d_B(\mathbf{x}, \mathbf{x}_i)}{\sigma_B} \right\|^2 \right) k \left(\left\| \frac{d_C(\mathbf{x}, \mathbf{x}_i)}{\sigma_C} \right\|^2 \right) k \left(\left\| \frac{l - l_i}{\sigma_l} \right\|^2 \right) \quad (2.13)$$

where $d_B(\mathbf{x}, \mathbf{x}_i)$ and $d_C(\mathbf{x}, \mathbf{x}_i)$ are obtained as in (2.8) and (2.9), and σ_B and σ_C are their bandwidths. To achieve invariance to illumination, a large bandwidth is applied to the brightness component so that the differences resulting from illumination changes can be given less weight.

2.4.2 Matching of Appearances

Given a human appearance, we need to find its best match in an appearance gallery already built. This requires to compute the similarity of the appearance to the appearance models in the gallery. We choose Kullback-Leibler distance (cross entropy) [17] to measure the similarity of two appearances. Kullback-Leibler distance measures the similarity of two distributions. Let $P(x)$ and $Q(x)$ be the distribution of two feature spaces, then their Kullback-Leibler distance is defined to be

$$D(P(x)||Q(x)) = \int_{-\infty}^{+\infty} P(x) \log \frac{P(x)}{Q(x)} dx \quad (2.14)$$

where $D(P(x)||Q(x)) \geq 0$ and equality holds when the two distributions are identical. In the following we will discuss in the detail how Kullback-Leibler distance is applied in appearance matching, especially in the above proposed appearance model with kernel density estimation.

Suppose in the gallery there are n appearance models built from n snapshots $S^{(J)} (J = 1, 2, 3, \dots, n)$ of n different appearances, and that the distribution $p^{(J)}(\mathbf{x}, l)$ of the feature space $(\mathbf{X}, \mathbf{L})^{(J)}$ of the snapshot of appearance J is obtained as described in section 2.4.1. Then the similarity of the model distribution $p^{(J)}(\mathbf{x}, l)$ and the distribution $p^{(K)}(\mathbf{x}, l)$ of a test image K can be measured by their Kullback-Leibler distance

$$D(p^{(K)} || p^{(J)}) = \int p^{(K)}(\mathbf{x}, l) \log \frac{p^{(K)}(\mathbf{x}, l)}{p^{(J)}(\mathbf{x}, l)} d\mathbf{x} dl \quad (2.15)$$

Formulae (2.15) can be rewritten as follows

$$\begin{aligned} D(p^{(K)} || p^{(J)}) &= - \int p^{(K)}(\mathbf{x}, l) \log p^{(J)}(\mathbf{x}, l) d\mathbf{x} dl \\ &\quad - \left(- \int p^{(K)}(\mathbf{x}, l) \log p^{(K)}(\mathbf{x}, l) d\mathbf{x} dl \right) \end{aligned} \quad (2.16)$$

The first term in (2.16) measures how unexpected the feature space $(\mathbf{X}, \mathbf{L})^{(K)}$ of distribution $p^{(K)}(\mathbf{x}, l)$ was from the model distribution $p^{(J)}(\mathbf{x}, l)$, and the second term measures how unexpected $(\mathbf{X}, \mathbf{L})^{(K)}$ is from the true distribution. So (2.16) is also a measure of information gain [9]. K is a true correspondence with J , or snapshot K and snapshot J are of the same person, if the unexpectedness of $(\mathbf{X}, \mathbf{L})^{(K)}$ from model $p^{(J)}(\mathbf{x}, l)$ is minimized or the Kullback-Leibler distance is minimized, that is,

$$J = \arg \min_{J \in S} D(p^{(K)} || p^{(J)}) \quad (2.17)$$

where S is the set $S = \{S^{(J)}, J = 1, 2, \dots, n\}$ of the snapshots in the database. Here we should note that in (2.15) the integral is over the entire feature space. We can

rewrite (2.15) as

$$D\left(p^{(K)}||p^{(J)}\right)=E_{(\mathbf{X},\mathbf{L})^{(K)}}\left[\log\frac{p^{(K)}(\mathbf{x},l)}{p^{(J)}(\mathbf{x},l)}\right] \quad (2.18)$$

which reflects the fact that the Kullback-Leibler distance is an average log-likelihood ratio over the feature space $(\mathbf{X},\mathbf{L})^{(K)}$. In practice we have the sample feature values of the pixels inside the silhouette. From the weak law of large numbers

$$D\left(p^{(K)}||p^{(J)}\right)=\frac{1}{N^{(K)}}\sum_{i=1}^{N^{(K)}}\log\frac{p^{(K)}(\mathbf{x}_i,l_i)}{p^{(J)}(\mathbf{x}_i,l_i)} \quad (2.19)$$

with probability 1, where (\mathbf{x}_i,l_i) are the sample feature values of the pixels from the snapshot of appearance K . Here we should note that it is not necessary to use all the pixels inside the silhouette to calculate (2.19). As long as the number of samples is large enough, (2.19) is true with probability 1. So, we can sample the pixels inside the silhouette and average the log-likelihood ratio, saving significant computation. The samples should be chosen according to the distribution of path-length so that they are evenly distributed over the whole silhouette. To achieve that, the pixels are first ranked in order of path-length, then the sampling is performed so that more samples are taken at path-lengths with larger probabilities.

In (2.19) both $p^{(J)}(\mathbf{x},l)$ and $p^{(K)}(\mathbf{x},l)$ are derived by kernel density estimation using the sample pixel values of the image in the respective snapshot. They can be expressed as follows

$$p^{(J)}(\mathbf{x}_i,l_i^{(K)})=\frac{1}{N^{(J)}}\sum_{m=1}^{N^{(J)}}k\left(\left\|\frac{\mathbf{x}_i^{(K)}-\mathbf{x}_m^{(J)}}{\mathbf{h}_{\mathbf{x}}}\right\|^2\right)k\left(\left\|\frac{l_i^{(K)}-l_m^{(J)}}{\sigma_l}\right\|^2\right) \quad (2.20)$$

$h_r = h_g = h_b$	10	10	10	10	10	15	15
σ_l	0.01	0.02	0.05	0.1	0.2	0.01	0.02
$P_I + P_{II}$	85.71	83.04	90.63	90.63	91.07	79.46	73.66
P_I	12.5	25	25	43.75	0	25	25
P_{II}	73.21	58.04	65.63	46.88	91.07	54.46	48.66
$h_r = h_g = h_b$	15	15	15	20	20	20	20
σ_l	0.05	0.1	0.2	0.01	0.02	0.05	0.1
$P_I + P_{II}$	87.05	89.29	90.63	78.13	81.7	82.59	88.84
P_I	25	37.5	0	18.75	12.5	0	0
P_{II}	62.05	51.79	90.63	59.38	69.2	82.59	88.84

Table 2.2: Bandwidth selection for R,G,B color and pathlength

$$p^{(K)}(\mathbf{x}_i^{(K)}, l_i^{(K)}) = \frac{1}{N^{(K)}} \sum_{m=1}^{N^{(K)}} k \left(\left\| \frac{\mathbf{x}_i^{(K)} - \mathbf{x}_m^{(K)}}{\mathbf{h}_{\mathbf{x}}} \right\|^2 \right) k \left(\left\| \frac{l_i^{(K)} - l_m^{(K)}}{\sigma_l} \right\|^2 \right) \quad (2.21)$$

where the superscript indicates from which person the sample feature value is drawn. So formulae (2.20) provides the probability of feature value $(\mathbf{x}_i^{(K)}, l_i^{(K)})$ appearing in the feature space of appearance J , and formulae (2.21) provides the probability of feature value $(\mathbf{x}_i^{(K)}, l_i^{(K)})$ appearing in the feature space of appearance K .

2.5 Evaluation

In this section we will evaluate the proposed appearance model. We will first study the influence of bandwidth on appearance matching and the optimal bandwidth is selected. Then the appearance model is evaluated and effectiveness of the proposed appearance model is demonstrated.

h_{Bright}	10	10	10	10	20	20	20	20
h_{Color}	3	3	5	5	1	1	1	3
σ_l	0.02	0.05	0.02	0.05	0.02	0.05	0.1	0.02
$P_I + P_{II}$	68.75	75.89	69.2	75.89	34.38	42.86	50.89	45.09
P_I	37.5	31.25	18.75	18.75	12.5	12.5	12.5	18.75
P_{II}	31.25	44.64	50.45	57.14	21.88	30.36	38.39	26.34
h_{Bright}	20	20	20	20	20	35	35	35
h_{Color}	3	3	5	5	5	1	1	1
σ_l	0.05	0.1	0.02	0.05	0.1	0.01	0.02	0.05
$P_I + P_{II}$	53.13	56.7	51.79	59.82	63.39	23.21	25.45	33.48
P_I	6.25	6.25	18.75	31.25	6.25	6.25	0	0
P_{II}	46.88	50.45	33.04	28.57	57.14	16.96	25.45	33.48
h_{Bright}	35	35	35	35	35	35	35	50
h_{Color}	1	3	3	3	5	5	5	1
σ_l	0.1	0.02	0.05	0.1	0.02	0.05	0.1	0.01
$P_I + P_{II}$	39.73	28.57	34.82	40.18	31.7	36.61	39.29	18.75
P_I	0	0	6.25	6.25	6.25	6.25	6.25	0
P_{II}	39.73	28.57	28.57	33.93	25.45	30.36	33.04	18.75
h_{Bright}	50	50	50	50	50	50	50	50
h_{Color}	1	1	1	3	3	3	5	5
σ_l	0.02	0.05	0.1	0.02	0.05	0.1	0.02	0.05
$P_I + P_{II}$	26.79	33.93	40.18	20.98	27.23	30.36	21.88	29.46
P_I	0	0	0	0	0	0	0	0
P_{II}	26.79	33.93	40.18	20.98	27.23	30.36	21.88	29.46

Table 2.3: Bandwidth selection for brightness and path-length

2.5.1 Bandwidth Selection

Bandwidth selection has been studied under different scenery. In [23], the optimal bandwidth is the one that minimize the mean integrated squared error, which is of little practical use since it depends on the estimate of the Laplacian of the unknown density. In [2], a bandwidth selection algorithm is proposed to maximize the mean shift vector to fit the scale of the underlying data, which is useful for segmentation applications. Here our bandwidth selection is for getting the optimal matching rate. We hope that the bandwidth should both give good resolution to discriminate difference in feature and tolerate some changes brought by factors such as brightness.

We assume that we have some training sequences before we do the actually matching, which is possible in most cases. From these training sequences we can calculate the Kullback-Leibler distances between images of both the same person and different person. So we have two sets of distances, the distance between images of the same person D_s and the distance between images of different person D_d . From these two sets of distances, we can calculate two types of error, type I error is the error of taking the same person as different and type II error is the error of taking different person as the same. These two types of error can be calculated respectively as

$$P_I(\mathbf{h}_\mathbf{x}, \sigma_l, T) = \frac{\# \text{ of } D_s > T}{\text{total } \# \text{ of } D_s} \quad (2.22)$$

$$P_{II}(\mathbf{h}_\mathbf{x}, \sigma_l, T) = \frac{\# \text{ of } D_d \leq T}{\text{total } \# \text{ of } D_d} \quad (2.23)$$

where T is a threshold. By varying T , we can get different type I and type II error. We hope that both types of error should be small. So the optimal bandwidth should minimize both type I and type II error

$$(\mathbf{h}_{\mathbf{X}}, \sigma_l) = \arg \min_{\mathbf{h}_{\mathbf{X}}, \sigma_l} \min_T P_I + P_{II} \quad (2.24)$$

Table 2.2 and table 2.3 show type I and type II error at different bandwidth when different features are used. In table 2.2 path-length and R, G, B values are used for kernel estimation. Table 2.3 employs the feature of path-length and brightness and color which are obtained from R, G, B values as described in section 2. In both tables we use two sets of images of eight people. The two sets of images are taken from two different places by two different cameras. From table 2.2 we can see that when $h_r = h_g = h_b = 15$ and $\sigma_l = 0.02$ both types of error are small. Table 2.3 shows that the optimal bandwidth should be $h_{brightness} = 50$, $h_{color} = 1$, and $\sigma_l = 0.01$. Here the optimal brightness bandwidth is very large which make the scheme tolerate the brightness change under different cameras. To save the space, we will not list the two types of errors of all the features discussed in section 2.3.2. We just show the resulted optimal bandwidth in table 2.4. In the following experiments without specific indication, the bandwidth of features will be set as in table 2.4.



(a) Image sequences taken by camera 1



(b) Image sequences taken by camera 2

Figure 2.2: Dataset Honeywell: Indoor image sequences taken by two cameras collected by the Honeywell corporation

Feature	Bandwidth
RGB and path-length	$h_R = h_G = h_B = 15, \sigma_l = 0.02$
RGB rank and path-length	$h_R = h_G = h_B = 3, \sigma_l = 0.02$
Brightness,color and path-length	$\sigma_B = 50, \sigma_C = 1, \sigma_l = 0.01$
Normalized RGB	$h_r = h_g = 0.02, h_b = 20$

Table 2.4: Optimal Bandwidth of different features

2.5.2 Experiment Setting

In our experiment, codebook based background subtraction [16] was first used to segment the moving people. Then small noise is filtered and morphological operations of closings and connected component analysis are used to obtain the silhouettes of people.

The data set was collected by the Honeywell corporation. Two indoor cameras captured 30 appearances, each under different lighting conditions, from different locations and with people moving in different directions with respect to the cameras. Figure 2.2(a) and figure 2.2(b) show example images of the thirty different appearances taken by the two cameras. Although some appearances are actually from the same person, they are clothed differently and are treated as different in our experiments.

In our experiment, we will study the performance of different features, the influence of illumination and pose to them, the effects of scale and sampling. Finally one

Feature used	Camera 1 matches with camera 2	Camera 2 matches with camera 1
RGB and path-length	13.33%	23.33%
RGB rank and path-length	86.67%	100%
Brightness, color and path-length	76.67%	73.33%
Normalized RGB and path-length	20%	40%
RGB rank only	66.67%	63.33%

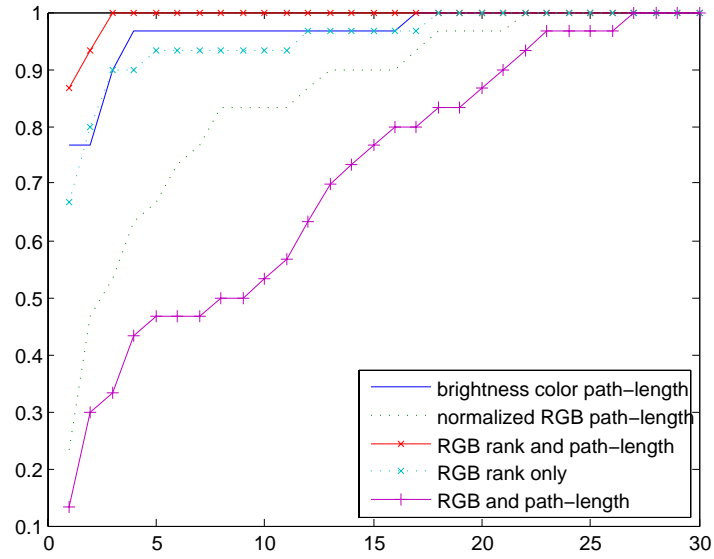
Table 2.5: Performances of different features when snapshots of the two cameras are of similar pose

possible application, detection of local appearance difference or change is discussed.

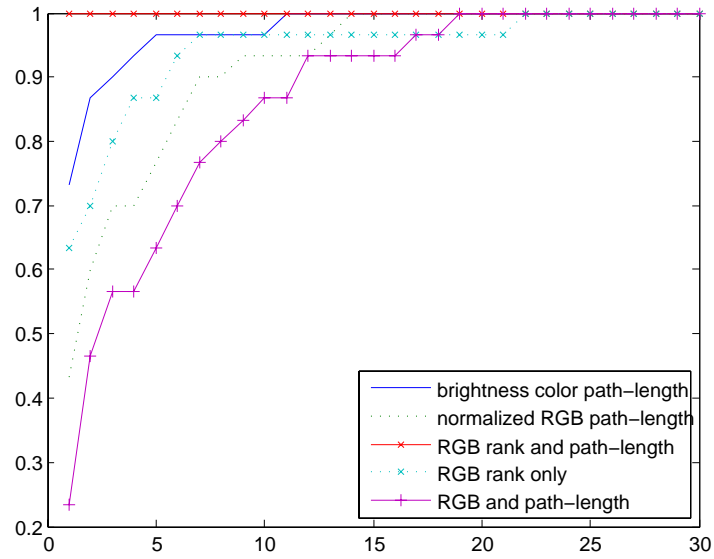
2.5.3 Study of Color-Pathlength Profile

We first manually selected snapshots from the video sequences so that the appearances have similar poses in the two cameras with good background subtracted silhouettes. So, the two sets of data have the same pose and different illumination conditions. We evaluate different features including RGB, normalized RGB, brightness and color and path-length, and RGB rank with and without path-length as shown in table 2.5. The cumulative match curves (CMC) are shown in figure 2.3.

From table 2.5 and figure 2.3 we see that the performance of the RGB features are the worst and RGB rank the best. This is because the sequences are taken indoors



(a) CMC of different features when camera 1 matches with camera 2



(b) CMC of different features when camera 2 matches with camera 1

Figure 2.3: CMC of different features when snapshots of the two cameras are of similar pose

where the illumination varies significantly as people move relative to an extended and proximal light source. This is also reflected in the optimal bandwidths for the brightness and color features, which are determined to be $\sigma_B = 50$, $\sigma_C = 1$. This means that the best classification performance used a large brightness bandwidth to ignore the severe brightness changes between cameras. However large brightness bandwidths are problematic since sometimes a bright color may be confused with a dark color. Figure 2.4 shows an example of mismatching when brightness, color and path-length features are used and a bright color is confused with a dark color. That is why the feature of brightness color and path-length is not so discriminative as that of RGB rank and path-length. As RGB rank together with path-length is the most invariant to illumination, in the following experiments we will usually only show performance with the feature of RGB rank and path-length.

Table 2.5 and figure 2.3 also show that if only RGB rank feature is used, the performance drops significantly compared with that with the path-length feature, illustrating the importance of path-length in discriminating different appearances.

Next, we manually selected two snapshots of each appearance from camera 1 so that the snapshots are taken approximately at the same place but with different poses. In this way, we have two sets of appearances that are of the same illumination and different pose. Also we randomly picked snapshots from the sequences of the two cameras so that we have snapshots of different illumination and different pose. All the above snapshots have good background subtraction result. Table 2.6 shows



Figure 2.4: Using a large brightness bandwidth may confuse bright color with dark color

the matching result of the three sets of snapshots. From the result it is observed that when illumination invariant color features are used, the influence of pose on the matching result is marginal. This illustrates the pose insensitivity of the path-length feature. In the above match, all the snapshots are selected manually with good background subtraction result. We also perform matching when the snapshots are randomly selected from video sequences without knowing if the silhouette is good or not. The matching result is also shown in table 2.6. The matching rate drops to 70% and 76.67% when the silhouettes of snapshots are not necessarily good. That means simply matching snapshots are not enough to get correct matches and we have to use the information in the sequence, which will be discussed in the next chapter.

In summary the above experiments show that RGB rank is the most illumination invariant in the color features discussed. Path-length is indispensable to achieve good matching result. By using path-length feature, the influence of pose difference

Sequence set	Camera 1 matches with camera 2	Camera 2 matches with camera 1
Different illumination same pose	86.67%	100%
Same illumination different pose	93.33%	96.67%
Different illumination different pose	93.33%	86.67%
Snapshots with possible bad silhouettes	70%	76.67%

Table 2.6: Performance of different sequence sets. Feature of RGB rank and path-length is used

is negligible.

2.5.4 Study of Scale and Subsampling

To study the performance when snapshots of different size are matched, we filtered the snapshots of camera 1 and scaled each dimension to $1/2$, $1/4$, $1/8$ of the original size, then matched them with the snapshots of camera 2. Table 2.7 shows that only when the size is scaled to $1/8$, does the matching rate drop significantly, which demonstrate the scale invariance of the proposed model.

We also studied the influence of sampling as mentioned in section 2.4.2. Table 2.8 shows that when one fourth of the pixels are used the performance slightly degrades. When only 500 pixels are selected, the performance even improves a little compared with using one fourth of the pixels. This is partly because when fewer pixels are sampled, the chance of selecting the noisy pixels due to imperfect segmentation is

Sequence set	Camera 1 matches with camera 2	Camera 2 matches with camera 1
Original Scale	93.33%	86.67%
Snapshots of camera 1 scaled by 1/2 in each dimension	93.33%	86.67%
Snapshots of camera 1 scaled by 1/4 in each dimension	93.33%	76.67%
Snapshots of camera 1 scaled by 1/8 in each dimension	86.67%	63.33%

Table 2.7: Performance of matching when snapshots of camera 1 are scaled. Feature of RGB rank and path-length is used

reduced. When all pixels are used, it takes about 2.5 hours to conduct the two-way matching (Intel XEON CPU1.8GHz RAM1GMB). If one-fourth of the pixels are sampled, only about 1 hour is used for the two-way matching. If 500 pixels are sampled, the time for matching is reduced to about 25 minutes. So by sub-sampling the image, the time for computation can be reduced significantly with little sacrifice in matching accuracy.¹ So although our algorithm seems to be demanding in computation, by subsampling the computation can be greatly reduced without sacrificing too much performance.

¹Here in the calculation we did not do any optimization of the program and did not employ recently developed efficient algorithms of fast calculation of kernel density estimation [7], which would further improve the speed of modeling and matching.

Samples used	Camera 1 matches with camera 2	Camera 2 matches with camera 1
All pixels are used	86.67%	100%
One-fourth of the pixels are used	83.33%	86.67%
500 pixels are used	90%	90%

Table 2.8: Matching result of the sequence set of different illumination same pose when sub-sampling is applied. Feature of RGB rank and path-length is used

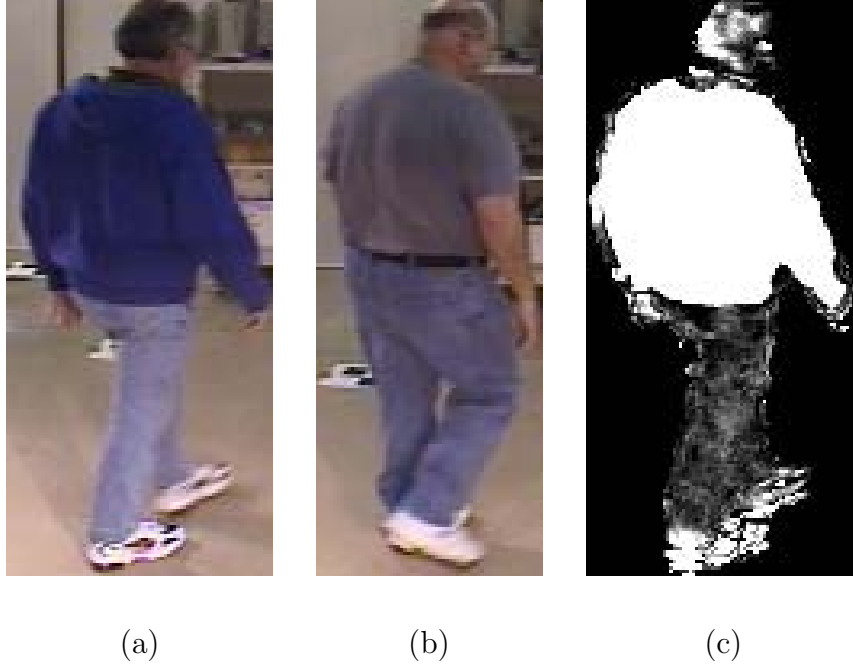


Figure 2.5: The log-likelihood ratio image (c) reflects the local appearance difference between the test image (a) and the model image (b). Here a brighter pixel indicates a larger log-likelihood ratio

2.5.5 Local Appearance Difference

Another interesting observation is that if we compute the pixel based log-likelihood ratio image as shown in figure 2.5, where the brighter a pixel is the greater the log-likelihood ratio, we can observe that the image reflects appearance differences. In figure 2.5 the two people wear different jackets but similar blue jeans, so the log-likelihood ratio in the upper part of the body is very large, but very small in the lower part of the body. So, by analyzing the log-likelihood ratio image, we can detect local appearance differences.

2.6 Conclusion

We have proposed our statistical human appearance model in this chapter. The appearance model is based on color path-length profile. Each pixel in the silhouette of human appearance is represented by the feature of color and path-length. The color feature can be ordinary RGB color, or illumination invariant color feature such as normalized RGB, RGB rank or brightness and color decomposition of RGB space. Path-length not only describes the structural property of each pixel but also achieves invariance to pose and motion of human body. Our model is nonparametric, that is, it uses kernel density estimation to get the density of color path-length feature. Kullback-Leibler distance measures the distance of an example appearance to the models in the appearance gallery. Our experiment results demonstrate the effectiveness of the proposed human appearance model. Experiment result also show that the feature of RGB rank and path-length is the most discriminative among all

the color path-length features and the importance of path-length in the proposed appearance model.

Chapter 3

Human Appearance Modeling of Video Sequence

3.1 Overview

In the last chapter we have proposed the human appearance model built from snapshot. The experiments there have shown the effectiveness of the proposed appearance model when the snapshots are selected with good silhouettes. When the silhouettes in the snapshots are noisy, or background subtraction produces noisy results, which happens quite often, the matching rate deteriorates significantly. So matching with just one snapshot is not robust enough. In addition, the color path-length feature of one snapshot does not contain all the appearance information in the sequence. For example, when people walk their hands may move and occlude their torsos, which can change the color path-length features. Additionally a person may turn around, and new features may appear. So only using appearance features of one snapshot will result in mismatching. One solution is brute force solution, that is, to use all the frames in the tracks and then do all-to-all matching. However, this would require significant storage and computation, and would not take advantages of the redundancies among frames in video sequences.

In this chapter we try to build human appearance model from video sequence so that the model contains as much appearance information as possible and at the

same time the representation of the model is as compact as possible. And ideally the computation involved is as small as possible. This chapter will propose two schemes of constructing appearance model from video sequence. In the first scheme a set of key frames are selected to represent the appearance in the sequence. In the second scheme similar appearances are clustered to get several appearance models from the sequence.

3.2 Key Frame Selection

3.2.1 Algorithm

In this section, multiple key frames are selected from the video sequence to represent the appearance in the sequence. In video sequences, there are a lot of redundancies among frames. So it is not necessary to use the appearances in all the frames to build the appearance model. We only need to select those frames that contains major appearance changes, for example, new features appear, or large pose change that leads to great color path-length change, which means that the appearance changes will result in a large information gain. So selecting key frames is equal to detecting changes of information gain, or Kullback-Leibler distance.

Suppose for appearance J we have one track $T^{(J)}$ containing M consecutive images $T^{(J)} = \{I_1^{(J)}, I_2^{(J)}, \dots, I_M^{(J)}\}$. The key frame selection process is as follows. The first frame is selected as the first key frame; it becomes the “current key frame”

K_i ($i = 1$ for the first key frame) for the following steps. Then, we calculate the Kullback-Leibler distance of the subsequent frames to the current key frame. If the current Kullback-Leibler distance is greater than a threshold, the current frame becomes the “next current key frame” K_{i+1} . In this way, those frames with large information gain or having new information are selected, and those not selected can be explained by the key frames.

Considering that sometimes random noise may corrupt the silhouettes and destroy the temporal coherence of appearance change in the video sequence, we can do some post-processing to eliminate those key frames that can explain less than a predefined number (in our experiment in section 3.2.2, the number is set to 3) of video frames. In this way, some noise appearances can be deleted.

Once the key frames are selected, the distance L of two sequences I, J is defined as

$$L^{(I,J)} = \underset{i \in K^{(I)}}{\text{median}} \min_{j \in K^{(J)}} D(p_i^{(I)} || p_j^{(J)}) \quad (3.1)$$

where $K^{(I)}$ is the set of key frames of sequence I . So, for each key frame of sequence I the closest distance to the key frames of sequence J is first retrieved; then, we take the median of these closest distances to be the distance of the two sequences. If a sequence contains some poor segmentations that cannot be filtered out based on simple shape constraints, then the key frame selection may select outliers in the sequence as some of the key frames. So, computing the median of the closest distances of the key frames tends to eliminate the effect of these outliers.

Feature	Camera 1 matches with camera 2	Camera 2 matches with camera 1
RGB rank and path-length	93.33%	96.67%
Brightness, color and path-length	86.67%	73.33%

Table 3.1: Matching results of Honeywell sequences when the proposed key frame selection and matching scheme is used

3.2.2 Experiment Results

We studied performance when key frames are selected from video sequences as described in section 3.2.1. For each sequence of an appearance, where the length of the sequence varies from 15 frames to 30 frames, the key frames are picked as described in section 3.2.1. The threshold of one frame becoming a key frame is set to 2 when the RGB rank and path-length is used. When brightness color and path-length is used, the threshold is set to 2.5.

Table 3.1 demonstrates the matching results on the Honeywell dataset used in the snapshot matching in Chapter 2. Comparing table 3.1 to the last row of table 2.6 in chapter 2 we can see that by employing the proposed sequence matching method, compared with snapshot matching, significant improvement of matching rate is achieved. Figure 3.1 shows the key frames selected from one sequence with the sequence segment represented by each key frame and the corresponding Kullback-Leibler distances. In each image row the first image is the selected key frame. In

Feature	Camera 1 matches with camera 2	Camera 2 matches with camera 1
RGB rank and path-length	100%	100%
Brightness, color and path-length	91.67%	100%

Table 3.2: Matching results of outdoor sequences when the proposed key frame selection and matching scheme is used

the sequence of figure 3.1 we can see that some of the silhouettes are very bad, and the third key frame, which is corrupted due to segmentation error, is the only frame in that sequence segment (it actually can be filtered out in post-processing). By use of the robust distance measure between sequences, the effect of noisy key frames is reduced.

We also applied the key frame selection algorithm to a dataset that was taken outdoors, where there are twelve appearances with two different tracks for each appearance. One track was the sideview of a person walking from left to right and the other track was the sideview of the person walking from right to left. Example images are shown in figure 3.2. In this dataset, the threshold of one frame becoming a key frame is set to 1 and 2.5 respectively when RGB rank, path-length and the brightness color, path-length are used. Figure 3.3 shows the key frames of one sequence together with the sequence segment represented by each key frame and the corresponding Kullback-Leibler distances. From figure 3.3 we observe that in the sequence segment of key frame 1 the two legs of the person are almost together. Then in the next sequence segment, the two legs are largely separated. Finally































The key frame and its sequence segment						
						
Distance to the key frame	0	0.6647	1.3096	1.5268	1.7634	
<hr/>						
						
Distance to the key frame	0	0.9276	1.2262	0.8747	1.1314	
						
Distance to the key frame	1.2436	1.2809	1.6202			
<hr/>						
						
Distance to the key frame	0					
<hr/>						
						
Distance to the key frame	0	0.9897	1.1719	1.3345	1.2613	1.1787
						
Distance to the key frame	1.1249	1.2307	1.5913	1.2279	1.8880	1.1999
						
Distance to the key frame	1.1217	1.1197	0.9668	1.1354		
<hr/>						

Figure 3.1: Indoor sequence: example result of key frame selection



(a) sideviews of people walking from right to left



(b) sideviews of people walking from left to right

Figure 3.2: Dataset 2: Outdoor image sequences
















The key frame and its sequence segment					
					
Distance to the key frame	0	0.9423	0.9423		
<hr/>					
					
Distance to the key frame	0	0.7703	0.7959	0.6666	0.7286
					
Distance to the key frame	0.7094	0.7163	0.9101	0.8923	
<hr/>					
					
Distance to the key frame	0	0.7960	0.7967		
<hr/>					

Figure 3.3: Outdoor sequence: example result of key frame selection

in the last sequence segment one leg is bent which changes the color path-length profile. Here, in our sequences there are few view changes, so the key frames only demonstrate pose change. Table 3.2 shows the matching results when RGB rank, path-length feature and brightness, color and path-length feature are used.

3.3 Online Clustering

In section 3.2, key frame selection was proposed to construct the appearance model from video sequences. The key frames are selected by finding those frames with large information gain. However, using key frames to construct the appearance model has the following problems. First to get key frames, we have to select a threshold of information gain which is hard to decide. If the threshold is set too small, too many key frames would be produced, which makes matching algorithm very time consuming. If the threshold is set too large, not enough key frames would be produced, which leads to an appearance model not accurate enough and finally results in a lower matching rate. If we were to find an optimized threshold, some non-on-line algorithm must be used so that all the frames in the sequence are considered, which would lead to considerable computation. Another problem with the key frame selection algorithm is that the key frames are not compact enough. In most cases, people's pose and views recur. For example, at first the back view of person is to the camera and after some time the back view is again to the camera. In key frames selection, key frames are selected in adjacent frames, so those similar views in inconsecutive frames will be represented by key frames of similar content.

Thuse the key frames have overlapped information.

In this section and the next section, online hierarchical clustering is proposed to build the appearance model from video sequences. Since we do not know in advance the number of views or poses of a person may exhibit under a particular camera view, online clustering is first utilized to subtract the model for the representative views or poses in the video sequences. Online clustering aslo enables that clustering is performed on line as the data streams in so that there is no need to store all the patterns. The number of clustering depends greatly on the threshold for clustering. Here the strategy is to first use a relatively small threshold to do online clustering, which will lead to multiple appearance models for different views or poses, then hierachical clustering is used to further condense the model.

The orgnization of this section is as follows. First our online clustering scheme is discussed by presenting the traditional online clustering algorithm with an emphasis on the difference of our scheme. Then the effectiveness of the model is demonstrated by experiment results.

3.3.1 Algorithm

Table 3.3 shows a basic online clustering algorithm [4], i.e., a leader-follower clustering algorithm. As we can see from the table, an online clustering algorithm is composed of two steps. One is finding the nearest cluster, and the other is updating

the cluster center. We will elaborate these two steps in the following.

Table 3.3: A Classic Online Clustering Algorithm

Algorithm: Basic Leader-Follower Clustering Algorithm

```

w1 ← x

n ← 1

do

    accept new x

    j ← arg minj' dist(x, wj')

    if dis(x, wj') < θ

        then wj ← update(wj', x)

    else

        add new cluster wn+1 ← x

        n ← n + 1

until no more patterns

```

To find the nearest cluster, the distance of the new incoming data and the current cluster centers are to be computed. Here again Kullback-Leibler is used to measure the distance of the new data and the cluster centers. So the distance of x and \mathbf{w} is calculated as follows

$$D(x||\mathbf{w}) = \frac{1}{N^x} \sum_{i=1}^{N^x} \log \frac{p_x(x_i)}{p_{\mathbf{w}}(x_i)} \quad (3.2)$$

In (3.2), $x_i, i = 1, \dots, N^x$ are the samples from the current silhouette, $p_x(x_i)$ is the

probability density of x_i in the feature space of the current data, and $p_{\mathbf{w}}(x_i)$ is the probability density of x_i in the feature space of the cluster center \mathbf{w} , or it is the probability of appearance of the current data when the model is the current cluster center.

Usually the cluster is updated as a weighted average of the current cluster center and the current data. Here our scheme of cluster center updating cannot simply be the weighted average of the cluster center and the data. As we can see from the above explanation of (3.2), we use the model of the cluster center to predict the probability of the current data. So when the cluster center is updated with the current data, a revision of the model of the cluster is expected. As we use kernel density estimation to build the distribution model, or samples are used to compute the probability density of a specific feature value, the cluster center is updated by adding samples from the current data. We hope that every data or every silhouette plays an equal important role in the model of the cluster center. So a new cluster center is formed by the samples from the old cluster center and the samples of the new streamed-in silhouette. Suppose currently the cluster center is composed of samples from N silhouettes, then the cluster center is subsampled by $\frac{N}{N+1}$ and current new silhouette is subsampled by $\frac{1}{N+1}$ and the union of the samples from the two parts form the new cluster center. Table 3.5 shows the cluster center update algorithm. Here number of samples or sample size of the new cluster center is set to be the minimal of the sample size of the old cluster center and that of the incoming silhouette. Then the old cluster center and current incoming silhouette are sub-

sampled. The updated cluster center is the union of the subsampled sample sets.

Table 3.4: The Proposed Cluster Center Update Algorithm

Algorithm: Cluster Center Update Algorithm

Get new appearance model \mathbf{m} and get matched cluster center \mathbf{c}

$\mathbf{c}.\text{SampleSize} = \min(\mathbf{m}.\text{SampleSize}, \mathbf{c}.\text{SampleSize})$

$N = \mathbf{c}.\text{MemberNumber}$

$\text{SampleSet1} = \text{SubSample } \mathbf{m} \text{ by } \text{SampleSize} \times \frac{1}{N+1}$

$\text{SampleSet2} = \text{SubSample } \mathbf{c} \text{ by } \text{SampleSize} \times \frac{N}{N+1}$

$\mathbf{c}.\text{SampleSet} = \text{Union}(\text{SampleSet1}, \text{SampleSet2})$

$\mathbf{c}.\text{MemberNumber} = N + 1$

To save the time of computation, we make use of the fact that those appearances that are adjacent in time are similar and tend to be clustered into the same cluster. So cluster centers are put into a queue. Whenever a cluster center is updated, it is moved to the head of the queue so that when the next frame comes in it first matches with the head of the queue, or the most likely cluster center. If the Kullback-Leibler distance of the current appearance and the head of the cluster center queue is less than a threshold, which is quite likely due to the temporal correlation nature of video frames, the current appearance is clustered into the cluster and the cluster center should be updated. Otherwise, if the distance is greater than the threshold, those cluster centers in the queue are all matched. In this way, a lot of computa-

tion can be saved. If the distances of all the cluster centers are greater than the threshold, a new cluster is formed and added to the head of the queue. Table 3.5 summarizes the proposed online clustering algorithm.

Table 3.5: Proposed Online Clustering Algorithm

Algorithm: Online Clustering Algorithm

Get one NewAppearanceModel

if $Q.IsEmpty$

$Q.AddHead(NewAppearanceModel)$

elseif $KLdistance(Q.Head, NewAppearanceModel) < T$

$UpdateClusterCenter(Q.Head)$

else

$(MinimalKLDistance, BestMatchCluster) = Match(NewAppearanceModel, Q)$

if $MinimalKLDistance < T$

$UpdateClusterCenter(BestMatchCluster)$

$Q.MoveToHead(BestMatchCluster)$

else

$Q.AddHead(NewAppearanceModel)$

3.3.2 Experiment Results

Figure 3.4 and figure 3.5 show the online clustering results of two sequences. Let us denote the name of the sequence in figure 3.4 sequence A and the name of the sequence in figure 3.5 sequence B. From figure 3.4 and figure 3.5, we observe that those appearances of similar poses or views are clustered into the same cluster. For example, in figure 3.4, those frames of back view and front view are clustered into different clusters. cluster one two, three and six are the appearances of back views, and cluster four and five are the appearances of front views. For those back views of figure 3.4, in cluster one, the legs of the person are parted. In cluster two and three, the legs are crossed. And in cluster six, the legs are merged together. Another observation is that those appearances that are in the same cluster are not necessarily adjacent in time, which can be seen from the frame number. For example in cluster 1 of figure 3.4, silhouettes from frame 1 to 16, silhouettes from frame 34, 35 and silhouettes from frame 42 to 47 are in the same cluster. Similar observations can be acquired from figure 3.5. In the above examples, the threshold of online clustering is 1.5. Apparently if the threshold is larger, fewer clusters will be produced. So here we have the problem of deciding threshold. In the next section, we will discuss how to avoid the decision of threshold.

3.4 Online Hierarchical Clustering

In section 3.3, online clustering algorithm has been discussed. In the discussion, we have observed that the threshold of clustering plays an important role in






















































cluster 1										
	1	2	3	4	5	10	11	12	13	14
										
	15	16	34	35	42	43	44	45	46	47
cluster 2										
	6	7	8	9	36	37	38	39	40	41
cluster 3										
	17	18	48	49	50					
cluster 4										
	19	20	21	22	23	24	25	26	32	33
cluster 5										
	27	28	29	30	31					
cluster 6										
	51	52	53							

Figure 3.4: Online clustering result of sequence A




















































cluster 1									
	1	35	36	37	38	39	40	41	42
cluster 2									
	2	3	4	5	6	7			
cluster 3									
	8	9	10	11	12	13	14	15	16
									
	43	44	45	46	47				
cluster 4									
	17	18	19	48	49	50	51		
cluster 5									
	20	21	22	23	24	25	26	27	28
cluster 6									
	29	30	31	32	33	34			

Figure 3.5: Online clustering result of sequence B

clustering. At the same time, the threshold is not trivial to decide. If the threshold is too small, a lot of clusters are produced or a lot of sub-models for appearances are constructed. When appearances are matched, this will involve a lot of computation. If the threshold is set too large, few clusters are created. Then the appearance model is not descriptive enough, which will result a lot of mismatching. In this section, we will discuss how to avoid the selection of threshold. We propose online hierarchical clustering to solve the problem. Again, we will first discuss the algorithm, then experiment results are presented.

3.4.1 Algorithm

Hierarchical clustering is one of the best known methods in unsupervised learning. Given a set of data points, the output is a binary tree (dendrogram) whose leaves are the data points and whose internal nodes represent nested clusters of various sizes. The tree organizes these clusters hierarchically so that this hierarchy agrees with the intuitive organization of real-world data. Hierarchical structures are ubiquitous in the natural world. For example, the evolutionary tree of living organisms is a natural hierarchy. Here the intuition of using hierarchical clustering to construct appearance model is that when people move around, different views (front views, side views, or back views etc.) may be shown to the camera, under each view, there are different poses for example, sometimes the hands may be lifted up, or sometimes the hands may be inserted to the pockets etc. So human appearances have hierarchical structure.

Table 3.6: Agglomerative Hierarchical Clustering Algorithm

Algorithm: Agglomerative Hierarchical Clustering Algorithm [4]

initialize $m, \hat{m} \leftarrow n, D_i \leftarrow \mathbf{x}_i, i = 1, 2, \dots, n$

do $\hat{m} \leftarrow \hat{m} - 1$

 find nearest clusters, say D_i and D_j

 merge D_i and D_j

until $m = \hat{m}$

return m clusters

The classic method for hierarchically clustering data [4] is a bottom up agglomerative algorithm. It starts with each data point assigned to its own cluster and iteratively merges the two closest clusters together until all the data belongs to a single cluster. The nearest pair of clusters is chosen based on a given distance measure (e.g. Euclidean distance between cluster means, or distance between nearest points). Table 3.6 shows the agglomerative hierarchical clustering algorithm in [4], where m is the desired number of clusters, n is the initial number of points.

In traditional agglomerative clustering as in table 3.6, when two clusters are merged, usually their cluster members are simply pooled into one new cluster. In our case, each point or cluster to be merged is composed of samples from appearances. If samples from the clusters to be merged are simply pooled together, as the merge

Table 3.7: Algorithm of Merging Clusters

Algorithm: Merge Cluster c_1 and c_2 : Merge(c_1, c_2)

$c.\text{SampleSize} = \min(c_1.\text{SampleSize}, c_2.\text{SampleSize})$

$N_1 = c_1.\text{MemberNumber}$

$N_2 = c_2.\text{MemberNumber}$

$N = N_1 + N_2$

$\text{SampleSet1} = \text{SubSample } c_1 \text{ by SampleSize} \times \frac{N_1}{N}$

$\text{SampleSet2} = \text{SubSample } c_2 \text{ by SampleSize} \times \frac{N_2}{N}$

$c.\text{SampleSet} = \text{Union}(\text{SampleSet1}, \text{SampleSet2})$

$c.\text{MemberNumber} = N$

goes on, the number of samples in the cluster will become larger and larger. In reality, we do not need all the samples to build appearance models by kernel density estimation as we have discussed in chapter 2. So here a sampling method similar to the cluster center update scheme as shown in table 3.5 is used to merge clusters. Table 3.7 shows the algorithm of merging two clusters, where c_1 and c_2 are the clusters to be merged and c is the merged cluster.

In agglomerative clustering, we have to calculate $n(n - 1)$ interpoint distances, which is a significant computation. However, here the appearance models to be hierarchically clustered are built from video sequence, which has strong temporal correlation. Although the frames in a cluster are not necessarily temporally ad-

jacent, from the discussion of online clustering, we can see that they still exhibit temporal relation. So adjacent models tend to be similar. By making use of this fact, not all point-to-point or cluster-to-cluster distances are to be calculated. Only the distances to the neighboring points or clusters are computed. In this way, only $O(n)$ calculations are needed.

The agglomerative clustering algorithm provides no guidance in regard with which distance metric to choose. As the algorithm does not define a probability model of the data, usually Euclidean distance between means or distance between nearest points are empirically chosen. In [12], Bayesian hierarchical clustering algorithm is proposed, where marginal likelihoods are used to decide which clusters to merge. However, [12] assumes a parametric distribution for the data, which cannot describe complex data such as images. Here the proposed hierarchical clustering algorithm also uses the likelihood criterion to merge clusters, where the distribution models of the data are estimated using kernel density estimation.

Suppose D_i and D_j are two clusters, which are the clusters to be considered to be merged during hierarchical clustering. Before merging, the likelihoods $l_i(D_i)$ and $l_j(D_j)$ of D_i and D_j are as follows

$$l_i(D_i) = \frac{1}{N_i} \sum_{n=1}^{N_i} \log p^{(i)}(x_n^{(i)}) \quad (3.3)$$

$$l_j(D_j) = \frac{1}{N_j} \sum_{n=1}^{N_j} \log p^{(j)}(x_n^{(j)}) \quad (3.4)$$

where $x_n^{(i)}$ is the n -th sample of the model i which is also cluster D_i , or more precisely,

$D_i = \{x_n^{(i)}, n = 1, \dots, N_i\}$. $p^{(i)}(x_n^{(i)})$ is the probability density of $x_n(i)$ in the feature space of D_i and is obtained by kernel density estimation using the samples in the cluster of D_i

$$p^{(i)}(x_n^{(i)}) = \frac{1}{N_i} \sum_{m=1}^{N_i} K(x_n^{(i)} - x_m^{(i)}) \quad (3.5)$$

In fact, (3.3) and (3.4) are the entropy of cluster D_i and D_j respectively. If D_i and D_j are to be merged, and suppose D_k is the merged cluster and $D_k = \{x_n^k, n = 1, \dots, N_k\}$, the likelihoods $l_k(D_i)$ and $l_k(D_j)$ of D_i and D_j are

$$l_k(D_i) = \frac{1}{N_i} \sum_{n=1}^{N_i} \log p^{(k)}(x_n^{(i)}) \quad (3.6)$$

$$l_k(D_j) = \frac{1}{N_j} \sum_{n=1}^{N_j} \log p^{(k)}(x_n^{(j)}) \quad (3.7)$$

where $p^{(k)}(x_n^{(i)})$ is the probability density of $x_n(i)$ in the feature space of D_k and is obtained by kernel density estimation using the samples in the merged cluster D_k

$$p^{(k)}(x_n^{(i)}) = \frac{1}{N_k} \sum_{m=1}^{N_k} K(x_n^{(i)} - x_m^{(k)}) \quad (3.8)$$

$p^{(k)}(x_n^{(j)})$ is derived similarly by replacing index i in the equation (3.8) with index j . If D_i and D_j are merged, the decrease of likelihood is the least among all the candidate merges. The decrease of likelihood after merging is

$$\Delta l = l_k(D_i) + l_k(D_j) - l_i(D_i) - l_j(D_j) \quad (3.9)$$

$$= -\frac{1}{N_i} \sum_{n=1}^{N_i} \log \frac{p^{(i)}(x_n^{(i)})}{p^{(k)}(x_n^{(i)})} - \frac{1}{N_j} \sum_{n=1}^{N_j} \log \frac{p^{(j)}(x_n^{(j)})}{p^{(k)}(x_n^{(j)})} \quad (3.10)$$

$$= -D(D_i || D_k) - D(D_j || D_k) \quad (3.11)$$

The above equations tell us that the decrease of likelihood is the sum of the Kullback-Leibler distances of the unmerged clusters and the merged cluster.

Table 3.8 shows the detail steps of the proposed hierarchical clustering algorithm. In the algorithm, only neighboring merges are checked. A flag *checked* is introduced to avoid repeated calculation of Kullback-Leibler distances. So we do not need to calculate all the cluster-to-cluster distances at every level of merging. Only those distances of clusters that have been merged and resulted the likelihood change are computed.

In this section, we have proposed our hierarchical clustering algorithm. The cluster merging scheme and the computation of the distances between clusters are different from the traditional hierarchical clustering algorithm. Our appearance model is composed of feature samples from the appearances, so new clusters are formed by sampling. Unlike the traditional norm based distance computation, our merging criteria is according to the likelihood decrease of the data, which turns out to be the Kullback-Leibler distances between the merged cluster and the unmerged clusters. Those merges that lead to least likelihood decrease are combined to form the new cluster.

3.4.2 Experiment Results

In this section, we will demonstrate the experiment results of hierarchical clustering. Also matching results of appearance models based on hierarchical clustering are showed.

Figure 3.6 through figure 3.8 show the hierarchical clustering results of figure 3.4. Here those clusters with fewer than three images of human appearances are deleted because they are quite likely to be the noisy images. So cluster 6 in figure 3.4 is eliminated. Figure 3.9 through figure 3.12 show the hierarchical clustering results of figure 3.5. From these results, we can observe that by hierarchical clustering, those appearances of the same view are clustered into one cluster.

Once we have the appearance models obtained from hierarchical clustering, we can make use of the models to match appearances across cameras. The matching method is similar to the key frame matching. The distances $L_H^{(I,J)}$ of two sequences I, J of appearances are defined as

$$L_H^{(I,J)} = \text{median}_{i \in H^{(I)}} \min_{j \in H^{(J)}} D(p_i^{(I)} || p_j^{(J)}) \quad (3.12)$$

where $H^{(I)}$ and $H^{(J)}$ are the hierarchical clustering results of appearance I and J . So here we calculate the Kullback-Leibler distances between hierarchical clusters. We set the number of clusters to be 1, 2, 3, 4, and matches the appearances of camera 1 with those of camera 2 and then matches the appearances of camera 2 with those of camera 1. Table 3.9 shows the matching results, where RGB rank and path-length feature are used for appearance model construction. From table 3.9 it is observed that the matching rate is satisfactory. However, some of the rates are lower than that of key frame selection. This is because in key frame selection, many frames are selected as key frames, and the number of key frames are much more



















































cluster 6: merge of cluster 1, cluster 2	          1 2 3 4 5 10 11 12 13 14           15 16 34 35 42 43 44 45 46 47           6 7 8 9 36 37 38 39 40 41
cluster 3	     17 18 48 49 50
cluster 4	          19 20 21 22 23 24 25 26 32 33
cluster 5	     27 28 29 30 31

Figure 3.6: Level 1 of hierarchical clustering result of figure 3.4



















































cluster 6: merge of cluster 1, cluster 2										
	1	2	3	4	5	10	11	12	13	14
										
	15	16	34	35	42	43	44	45	46	47
										
	6	7	8	9	36	37	38	39	40	41
cluster 3										
	17	18	48	49	50					
cluster 7: cluster 4, cluster 5										
	19	20	21	22	23	24	25	26	32	33
										
	27	28	29	30	31					

Figure 3.7: Level 2 of hierarchical clustering result of figure 3.4





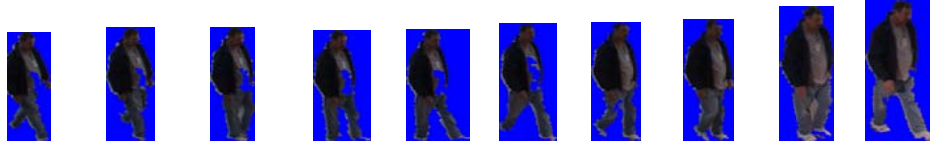

cluster 8: merge of cluster 6, cluster 3										
	1	2	3	4	5	10	11	12	13	14
										
	15	16	34	35	42	43	44	45	46	47
										
	6	7	8	9	36	37	38	39	40	41
cluster 7: cluster 4, cluster 5										
	17	18	48	49	50					
										
	19	20	21	22	23	24	25	26	32	33
										
	27	28	29	30	31					

Figure 3.8: Level 3 of hierarchical clustering result of figure 3.4




















































cluster 7: merge of cluster 1, cluster 2	        
	1 35 36 37 38 39 40 41 42
	     
	2 3 4 5 6 7
cluster 3	        
	8 9 10 11 12 13 14 15 16
	    
	43 44 45 46 47
cluster 4	      
	17 18 19 48 49 50 51
cluster 5	        
	20 21 22 23 24 25 26 27 28
cluster 6	     
	29 30 31 32 33 34

Figure 3.9: Level 1 of hierarchical clustering results of figure 3.5

















































cluster 7: merge of cluster 1, cluster 2	        1 35 36 37 38 39 40 41 42       2 3 4 5 6 7
cluster 8: merge of cluster 3, cluster 4	        8 9 10 11 12 13 14 15 16      43 44 45 46 47        17 18 19 48 49 50 51
cluster 5	        20 21 22 23 24 25 26 27 28
cluster 6	      29 30 31 32 33 34

Figure 3.10: Level 2 of hierarchical clustering results of figure 3.5

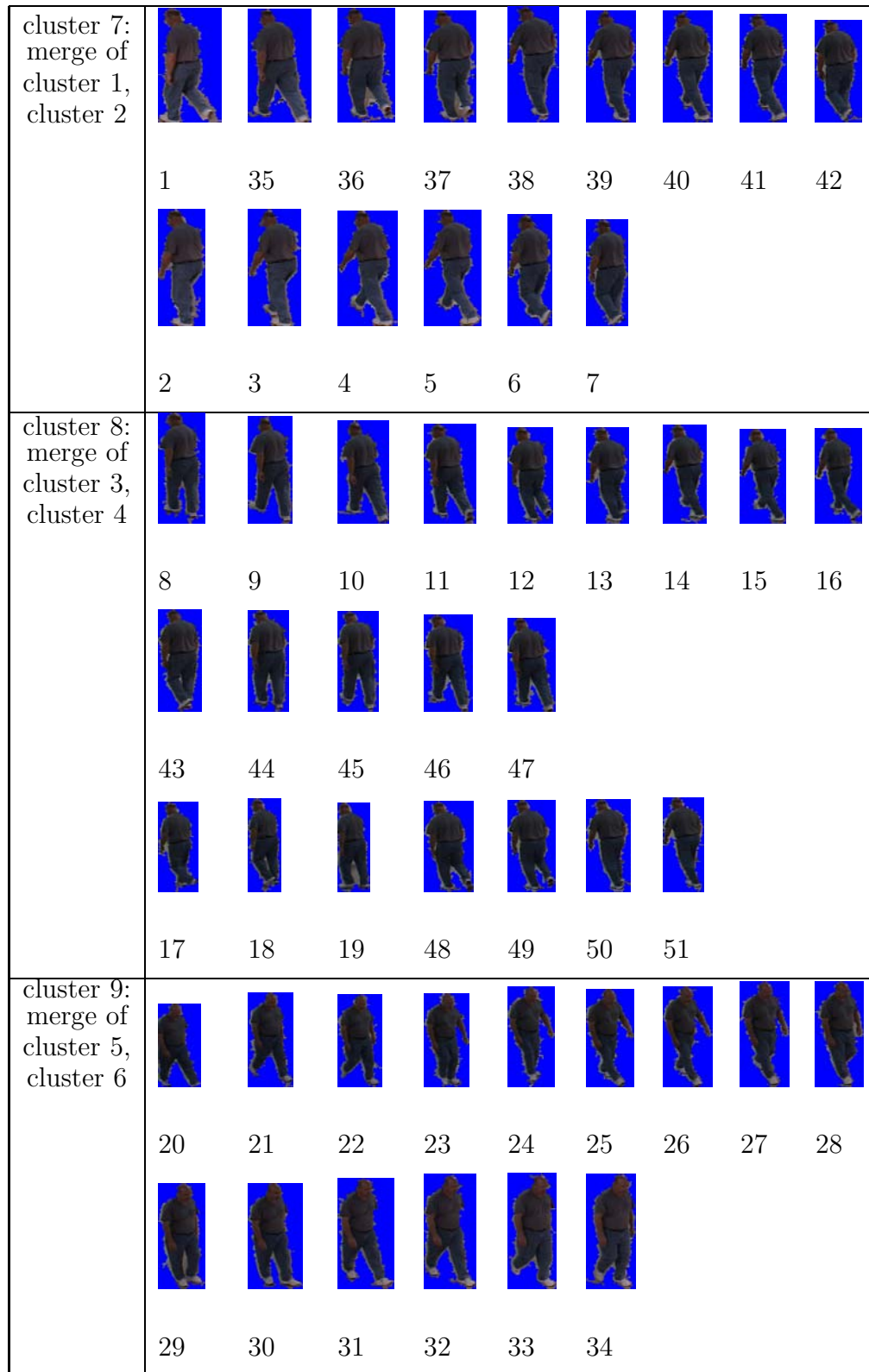


Figure 3.11: Level 3 of hierarchical clustering results of figure 3.5

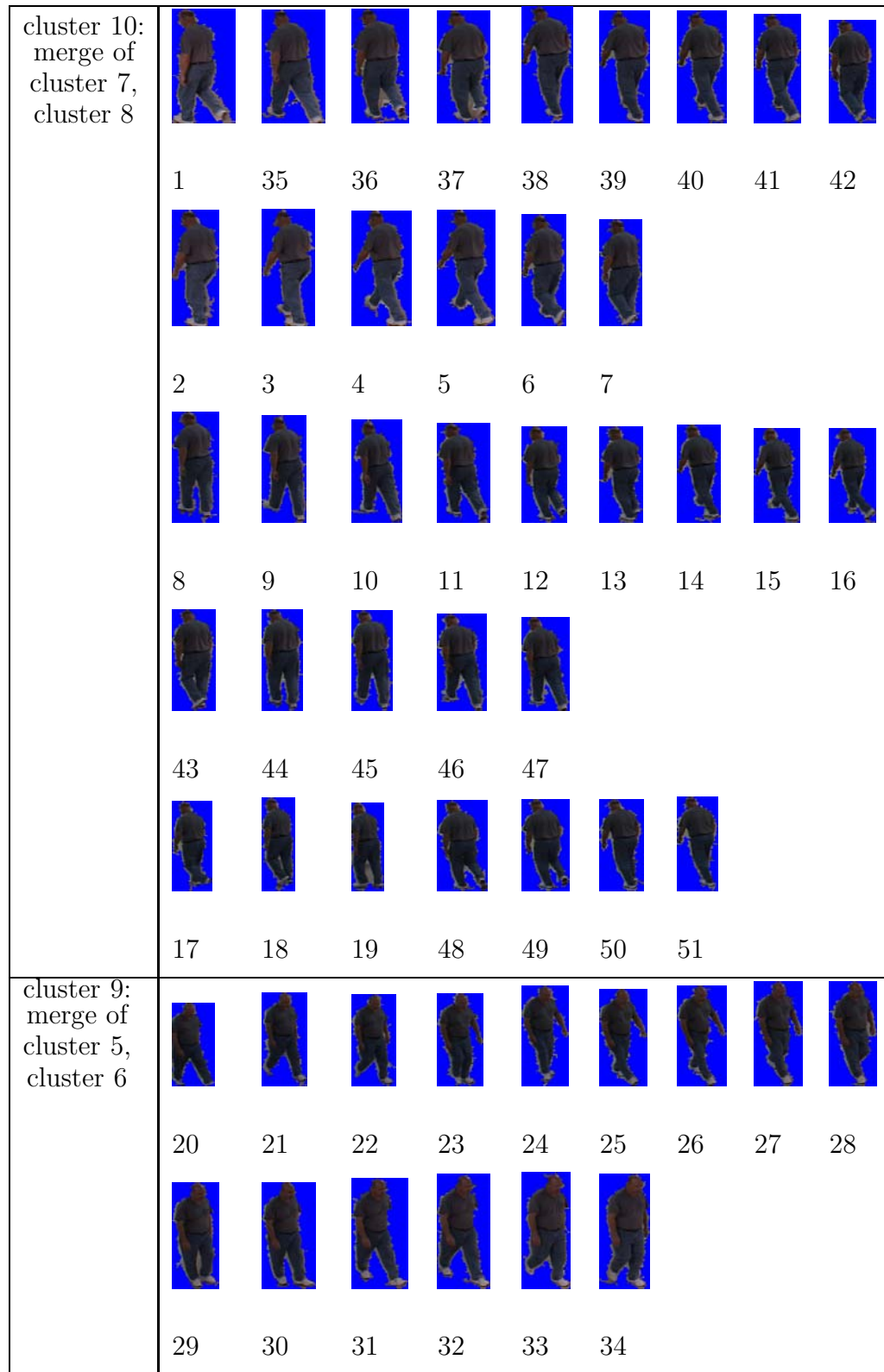


Figure 3.12: Level 4 of hierarchical clustering results of figure 3.5

than 4. Although key frame selection may give higher matching rate, it involves much more computation due to the large number of key frames.

3.5 Conclusion

In this chapter, construction of appearance model from video sequence is discussed. Two algorithms are proposed to build appearance models from video sequence, key frame selection and hierarchical clustering. In key frame selection, the key frames with large information gains are selected to represent the sequence. Then matching of appearances is achieved by matching those key frames. Experiment results show that key frame selection achieves very high matching rate. However, in key frame selection, the number of key frames is decided by the threshold. To achieve a high matching rate, a relatively small threshold is selected, which may lead to many key frames and involve significant computation in matching. Online hierarchical clustering algorithm reduces the number of sub-models by first online clustering the silhouettes and then hierarchically clustering the clusters obtained from online clustering. In this way, we only need to select a relatively small threshold for online clustering. Online hierarchical clustering can achieve high matching rate without significant computation in matching.

It is noted that although the online clustering and hierarchical clustering algorithms that are the components of our online hierarchical clustering are in the frame work of the classic algorithms, the cluster updating algorithm is different from traditional al-

gorithms. Here instead of some simple samples, the clusters are represented by a set of samples the model of which is estimated using kernel density estimation. So subsampling method is used to update clusters. Also since the cluster is represented by some samples, cluster distances cannot be directly calculated by traditional method. Likelihood of samples is calculated to decide if there should be a merge of cluster in hierarchical clustering, which turns out to be closely related with Kullback-Leibler distance. In fact, here we propose a hierarchical clustering algorithm when the model of the data to be clustered are estimated using non-parametric kernel density estimation. In such situation, when the clusters are to be merged, subsampling is performed. The distance of clusters is the average Kullback-Leibler distances of the individual clusters to the merged cluster.

Table 3.8: Hierarchical Clustering Algorithm for Appearance Modeling

Algorithm: Hierarchical Clustering Algorithm for Appearance Modeling

```

initialize  $m, \hat{m} \leftarrow n, D_i \leftarrow \mathbf{x}_i, i = 1, 2, \dots, n$ 

 $D_i.\text{checked} = 0;$ 

do  $\hat{m} \leftarrow \hat{m} - 1$ 

    for  $i = 1 : \hat{m}$ 

        if NOT  $D_i.\text{checked}$ 

             $D_{i,i+1} = \text{Merge}(D_i, D_{i+1})$ 

             $KLD_i = \text{KLDistance}(D_i, D_{i,i+1}) + \text{KLDistance}(D_{i+1}, D_{i,i+1})$ 

             $D_i.\text{checked} = 1$ 

        endif

    endfor

     $k = \arg \min_{1 \leq i \leq \hat{m}} D_{i,i+1}$ 

     $D_k = \text{Merge}(D_k, D_{k+1})$ 

     $\text{Remove}(D_{k+1})$ 

     $D_{k-1}.\text{checked} = 0$ 

     $D_k.\text{checked} = 0$ 

until  $m = \hat{m}$ 

return  $m$  clusters

```

Cluster number	1	2	3	4
camera 1 matches with camera 2	93.33%	93.33%	93.33%	93.33%
camera 2 matches with camera 1	83.33%	83.33%	86.67%	86.67%

Table 3.9: Matching results when appearance models are built using hierarchical clustering

Chapter 4

Conclusion

In this thesis, the problem of building correspondences of appearances taken at different places, at different times or across different cameras are explored. An appearance model is proposed to match images of appearances. Then the appearance model is further expanded to include information provided by video sequences. The contribution of the thesis can be summarized as follows.

First an appearance model based on color and path-length feature is proposed. To achieve invariance to illumination, the color feature of color rank and brightness-color is employed. Experiment results show that color rank feature and brightness-color feature is invariant to illumination changes compared with other commonly used color features. To represent the location of the color feature, path-length feature is used. Path-length of a pixel is the shortest path from a distinguished point, which we choose as the top of the head, to the pixel. Path-length provides structural information. It is also invariant to human motion and pose. Experiment results demonstrate that by adding path-length feature to the color feature, significant performance improvement is achieved. The probability model of appearance is constructed by kernel density estimation. Then Kullback-Leibler distance is utilized as the matching criteria. Experiment results give satisfactory performance of the

proposed appearance model and the matching scheme

To further improve the representability of the appearance model, the information in video sequence should be exploited. The first method extracts the information in video sequence by finding key frames. Key frames are those frames that have large information gains, or have large Kullback-Leibler distance to the previous key frame. Key frame selection method provides satisfactory matching results. However, threshold of key frame selection is hard to pick. The small threshold may result too many key frames and significant computation in matching, while a large threshold may lead to low matching rate.

To overcome the problem with key frame selection algorithm, online hierarchical clustering algorithm is proposed to formulate the appearance model from video sequence. The idea of online hierarchical clustering is that first online clustering algorithm is used to acquire the preliminary clusters by using a relatively small threshold. Then hierarchical clustering is applied to merge those very similar models. Unlike the traditional online clustering and hierarchical clustering algorithms, the data here are formed by a set of samples from the silhouettes and the model of the data cannot simply be represented by parametric model, instead nonparametric model is employed. So proper changes have to be made to traditional algorithms to incorporate these differences. Subsampling method is used to get the updated clusters. Likelihood change is computed for criteria of merging. Experiment results show satisfactory matching result even with few clusters.

Bibliography

- [1] Robert T. Collins and Ralph Gross and Jianbo Shi, “Silhouette-based human identification from body shape and gait,” *IEEE Conference on Automatic Face and Gesture Recognition*, 2002, pp. 351–356, May.
- [2] D. Comaniciu and V. Ramesh and P. Meer, “The Variable bandwidth mean shift and data-driven scale selection,” *Eighth Int’l Conf. Computer Vision*, 2001.
- [3] D. Comaniciu and V. Ramesh and P. Meer, “Kernel-based object tracking,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(5), pp. 564-577, 2003.
- [4] R. O. Duda and D. G. Stork and P. E. Hart, *Pattern Classification*, John Wiley & Sons, 2001
- [5] A. Elgammal and R. Duraiswami and D. Harwood and L. S. Davis, “Background and Foreground Modeling using Non-parametric Kernel Density Estimation for Visual Surveillance,” *Proceedings of the IEEE*, 90(7), pp. 1151-1163, 2002.
- [6] A. Elgammal, R. Duraiswami and L. S. Davis, “Probabilistic tracking in joint feature-spatial spaces,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

- [7] A. Elgammal, R. Duraiswami and L. S. Davis, “Efficient Kernel Density Estimation Using the Fast Gauss Transform for Computer Vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11), pp. 1499-1504, 2003.
- [8] Fieguth and D. Terzopoulos, “Color-based tracking of heads and other objects at video frame rates,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
- [9] J. A. Garcia and J. Valdivia and X. Vidal, “Information theoretic measure for visual target distinctness,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(4), pp. 362–383, 2001.
- [10] M. D. Grossberg and S. K. Nayar, “Determining the camera response from images: what is knowable?” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(11), pp. 1455-1467, 2003.
- [11] I. Haritaoglu and D. Harwood and L.S. Davis, “Ghost: a human body part labeling system using silhouettes,” *Proc. IEEE Fourteenth International Conference on Pattern Recognition*, 1998.
- [12] K.A. Heller and Z. Ghahramani, “Bayesian Hierarchical Clustering,” *In the Twenty-second International Conference on Machine Learning*, 2005.
- [13] O. Javed and Z. Rasheed and K. Shafique and M. Shah, “Tracking across multiple cameras with disjoint views,” *Tenth Int’l Conf. Computer Vision*, 2003.

- [14] O. Javed and K. Shafique and M. Shah, “Appearance modeling for tracking in multiple non-overlapping cameras,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [15] J. Kang and I. Cohen and G. Medioni, “Continuous tracking within and across camera streams,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
- [16] K. Kim and T.H. Chalidabhongse and D. Harwood and L.S. Davis, “Real-time foreground-background segmentation using codebook model,” *Real Time Img*, 11(3), pp. 172-185, June, 2005.
- [17] Solomon Kullback “Information theory and Statistics,” (Gloucester, Mass.: Peter Smith, 1978)
- [18] Jian Li and R. Chellappa, “Appearance Modeling Under Geometric Context,” *Tenth IEEE Int’l Conf. Computer Vision*, 2005.
- [19] Haibin Lin and David Jacobs, “Using the inner-distance for classification of articulated shapes,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [20] R. Nevatia, “A color edge detector and Its use in scene segmentation,” *IEEE Trans. Syst. Man, Cyb.*, 7(11), pp. 820-826.

- [21] Y. Shan and H.S. Sawhney and A. Pope, “Measuring the similarity of two image sequences,” *Asia Conference on Computer Vision*, 2004.
- [22] Y. Shan and H.S. Sawhney and R. Kumar, “Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [23] M. P. Wand and M. Jones, *Kernel smoothing* (Chapman and Hall, 1995)
- [24] O. Yamaguchi and K. Fukui and K. Maeda, “Face recognition using temporal image sequence,” *International Conference on Automatic Face and Gesture Recognition*, 1998.
- [25] C. Yang and R. Duraiswami and L. Davis, “Efficient mean-shift tracking via a new similarity measure,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.